

Преимущества нейронных сетей выражаются в их способности автоматически приобретать знания в ходе обучения, а также способности к обобщению, основной недостаток состоит в чувствительности к шуму во входных данных [2].

В итоге все алгоритмы интеллектуального анализа данных можно сравнить между собой, оценивая характеристики их свойств.

Как видно из данных табл. 1, каждый алгоритм имеет свои сильные и слабые стороны, но ни один метод не способен решить весь спектр задач интеллектуального анализа данных.

Список литературы

1. *Nefedov A.* Support Vector Machines: A Simple Tutorial // svmtutorial.online. URL: <https://svmtutorial.online/>.
2. *Шарабыров И. В.* Система обнаружения атак в локальных беспроводных вычислительных сетях на основе технологий интеллектуального анализа данных : дис. ... канд. тех. наук. Уфа, 2016. 144 с.
3. *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning. Springer, 2001.
4. Microsoft Decision Trees Algorithm // Электронный портал «Microsoft». URL: <https://docs.microsoft.com/en-us/sql/analysis-services/data-mining/microsoft-decision-trees-algorithm>.
5. *Уоссермен Ф.* Нейрокомпьютерная техника: теория и практика. М. : Мир, 1992.

УДК 004.056.53

Д. А. Корепин

Научный руководитель: д-р тех. наук, проф. С. В. Поршнева
Уральский федеральный университет, Екатеринбург

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ ВОССТАНОВЛЕНИЯ ИНФОРМАЦИИ В ФАЙЛОВОЙ СИСТЕМЕ REFS

Аннотация. В настоящей статье изучены и проанализированы официальные материалы по описанию файловой системы ReFS и механизмов ее работы. Рассмотрены возможности восстановления информации в данной файловой системе.

Ключевые слова: восстановление данных; файловая система; ReFS.

Файловая система ReFS была представлена в 2012 году [1]. С момента выпуска данной файловой системы прошло 5 лет, однако подробной документации

по ней все еще нет. Таким образом, тема исследования файловой системы ReFS сейчас является актуальной.

ReFS — сокращение от Resilient File System, что переводится как «Устойчивая файловая система». Такое название данная файловая система получила из-за концепций, заложенных при ее проектировании. По заявлению разработчика, ReFS является хорошо масштабируемой файловой системой для больших объемов данных, обеспечивающей высокую степень целостности данных [2].

В файловой системе ReFS для обеспечения отказоустойчивости используются такие технологии, как потоки целостности, зеркалирование, очистка мусора, проактивное исправление ошибок и прочие.

Для решения проблемы масштабируемости были разработаны структуры файловой системы, позволяющие хранить миллионы терабайт информации. Максимальный размер раздела, на котором можно развернуть ReFS, составляет 4,7 зеттабайта ($4,7 \cdot 10^{21}$ байт).

В качестве механизма хранения используются исключительно B+ деревья как единая структура для представления всей информации на диске. B+ дерево — структура данных на основе B-дерева, сбалансированное n -арное дерево поиска с переменным, но зачастую большим количеством потомков в узле. B+ дерево состоит из корня, внутренних узлов и листьев, корень может быть либо листом, либо узлом с двумя и более потомками [3].

Деревья могут быть встроены в другие деревья (корень дочернего дерева хранится в строке родительского дерева). Деревья могут быть очень большими и многоуровневыми или очень маленькими, включающими в себя всего несколько ключей и встроенными в другую структуру. Это обеспечивает чрезвычайно высокий уровень масштабируемости вверх и вниз во всех аспектах файловой системы. Наличие единой структуры значительно упрощает систему и уменьшает код. Новый интерфейс механизма включает в себя понятие «таблицы», которые являются множествами пар «ключ — значение». Большинство таблиц имеют уникальный идентификатор (называемый идентификатором объекта), с помощью которого они могут быть использованы. В системе присутствует специальная таблица, индексирующая все таблицы в системе. На рис. 1 представлена абстрактная структура ReFS.

Любое изменение информации в ReFS осуществляется в два этапа. На первом этапе создается новая измененная копия метаданных в свободном пространстве диска. На втором этапе производится обновление ссылки на новую копию. Такой механизм называется «копирование при записи» (Copy-on-Write) Благодаря этому можно обойтись без журналирования, автоматически сохраняя целостность данных [4].

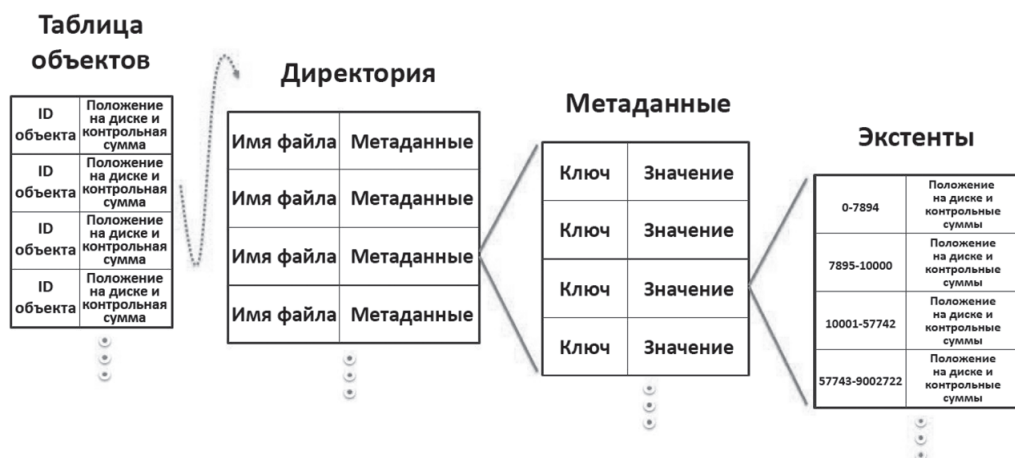


Рис. 1. Абстрактная структура данных в ReFS

Данная схема применима для содержимого файлов только в том случае, если при форматировании раздела была включена опция «проверка целостности на новом томе». При удалении файла производится перестроение структуры метаданных, сохраняя предыдущую версию дерева на диске. Это позволяет восстановить удаленные файлы до их перезаписи новыми данными.

Благодаря использованию технологии «копирование при записи», на разделе ReFS присутствует сразу же несколько копий файловых записей, что позволяет использовать это при восстановлении данных. В случае повреждения основной файловой записи можно восстановить одну из ее «старых» копий.

Технология CoW по умолчанию не распространяется на содержимое файлов, однако это можно включить, отформатировав раздел ReFS с помощью системной утилиты fsutil с параметром /i: enable. Этот параметр включает использование технологии Integrity Streams [2].

Кроме варианта восстановления копии MSB+, можно воспользоваться журналами файловой системы, которые располагаются в MLog записях. Содержание этих записей указывает на то, что в них содержится информация обо всех действиях в файловой системе (создание, изменение, удаление файлов и каталогов и т. д.).

На основании всего вышеперечисленного можно сделать вывод, что возможности восстановления информации в файловой системе ReFS шире, чем в других файловых системах Microsoft. После удаления файла файловая запись не удаляется и никакие изменения в нее не вносятся, что позволяет использовать информацию из этой записи для восстановления удаленного файла. Так как файловая запись становится неактуальной, она может быть перезаписана какой-либо другой информацией. Однако, кроме основной файловой записи, на разделе с большой долей вероятности можно найти копии этой файловой

записи (из-за применяемой технологии «копирование при записи»), которую также можно использовать для восстановления удаленной информации. При использовании Integrity Streams можно восстановить даже предыдущие версии файлов.

Список литературы

1. Sinofsky S. Building the next generation file system for Windows: ReFS // Блог разработки Windows 8 <https://blogs.msdn.microsoft.com/b8/2012/01/16/building-the-next-generation-file-system-for-windows-refs/>
2. Описание файловой системы ReFS // Документация Microsoft <https://docs.microsoft.com/en-us/windows-server/storage/refs/refs-overview>
3. В+ дерево // Национальная библиотека им. Н.Э. Баумана http://ru.bmstu.wiki/B%2B_%D0%B4%D0%B5%D1%80%D0%B5%D0%B2%D0%BE
4. Файловая система ReFS изнутри // R.LAB http://rlab.ru/doc/refs_file_system.html
5. ReFS integrity streams // Документация Microsoft <https://docs.microsoft.com/en-us/windows-server/storage/refs/integrity-streams>

УДК 629.519.248:056.57

А. А. Лазарева

Научный руководитель: проф. В. Е. Емельянов
Московский государственный технический университет
гражданской авиации, Москва

ОПРЕДЕЛЕНИЕ ОПТИМАЛЬНОГО ВРЕМЕНИ ПРОФИЛАКТИЧЕСКОГО ОБСЛУЖИВАНИЯ ПРОГРАММНО-АППАРАТНЫХ СРЕДСТВ

Аннотация. В данной работе рассматривается метод оценки времени проведения профилактических мероприятий программно-аппаратных средств. Показано, что при высокой надежности программного обеспечения и более низкой надежности аппаратной части для нахождения времени безотказной работы возможно использовать априорно известные плотности распределения вероятностей, и окончательная оценка формируется при нахождении искомого параметра по ПРВ, представляющей собой композицию возможных законов распределения. Результаты могут использоваться при определении гибких графиков проведения технического обслуживания, а также при переводе средств на техническую эксплуатацию по состоянию.